



第二次上机作业(程序控制)求解思路

说明:

- 作业可用多种方法求解,既可使用本文档提供的思路,也可按自己的想法设计程序;
- 注意练习独立调试程序。若程序有语法错误,请按照编译器“编译框”中的错误提示修改;若程序中有逻辑问题(运行结果与预想不同),可使用“单步调试”查找具体逻辑错误的语句,详见“超星学习通”1.2节视频最后部分。

注:  表示屏幕输出  表示键盘输入

基础编程题:

1. 输出0到100以内的所有素数之和。

提示:

(1) 素数指“大于1的自然数中,除了1和本身以外不再有其它因数的自然数”,因此在2到100的范围内(循环)找出素数即可;

(2) 可利用一个标志flag(整型)表示数据x是否为素数(如flag=1表示x为素数;flag=0表示x不是素数);

(3) 判断x是否为素数的一种方法:在2到x-1范围内取i(循环),若x%i等于0,则说明x非素数,可置flag=0,跳出循环;

(4) 若x的flag为1(素数),则累加。

2. 输出以下菱形图案(使用循环语句)。

```

*
***
*****
*****
*****
***
*

```

提示:

(1) 本题主要分析每行左边输出的“空格数”(可定义变量count1表示),及每行输出的“*数”(可定义变量count2表示),具体关系如图:

(2) 推导出count1、count2与i的关系后,外层循环控制i实现输出7行,内层有2个循环分别实现输出count1个空格、count2个*。

i (行数)	count1 (左空格数)	count2 (*数)
1	3	1
2	2	3
3	1	5
4	0	7
5	1	5
6	2	3
7	3	1

3. 输入一个任意正整数,求其中7出现的次数。

例: 输入 

输出 

提示:

(1) 可将输入的数据从低位开始每次取一位与7进行比较(从高位取更繁琐),取最低位数字可用“%”运算,如 $47758796\%10=6$,得最后一位6,与7进行比较;

(2) 对剩余的数字 $47758796/10=4775879$ 进行相同的操作,直至每一位均被取出并于7比较(循环)。

计算类编程题：4. 求 $1 + 1/2! + \dots + 1/n!$

输入正整数 n ($5 < n < 1000$)，计算上式前 n 项的和 (保留 4 位小数)。

例：输入：10

输出：1 + 1/2! + ... + 1/10! = 1.7183

提示：

- (1) 可用嵌套循环实现，外层循环用于求 n 项的和，内层循环用于求 $n!$ ；
- (2) 需要注意“1/非 1 整数=0”；
- (3) 注意输出格式。

5. 用二分法求方程 $2x^3 - 4x^2 + 3x - 6 = 0$ 在 (a, b) 之间的根。

例：输入 Please input the lower and upper boundaries: 3 10

输出 No root in this boundary!

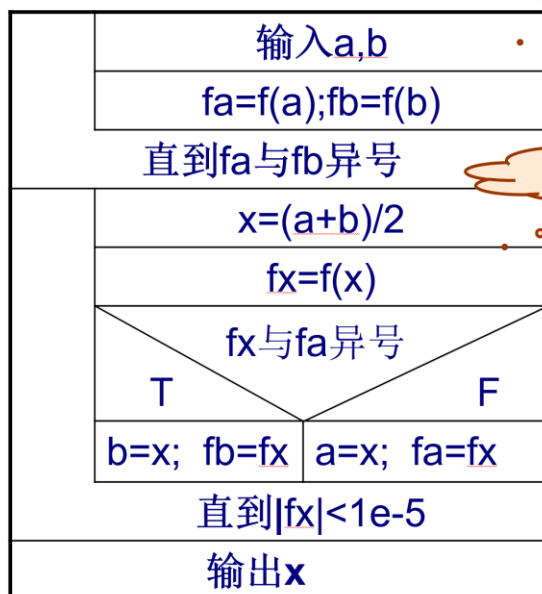
Please input the lower and upper boundaries: 1 5

The root is 2.00

提示：根据课堂提供思路求解。

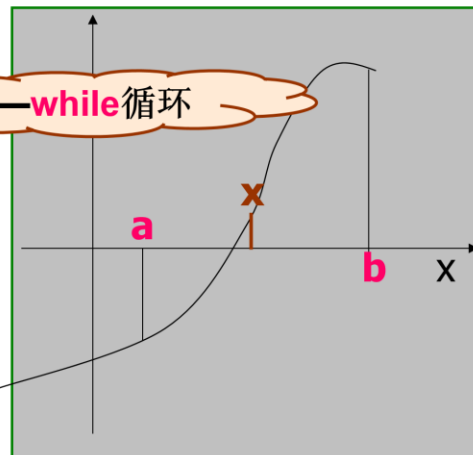
求方程 $2x^3 - 4x^2 + 3x - 6 = 0$ 的根，要求误差小于 10^{-5} 。

(2) 二分法



do-while 循环

do-while 循环



Back

设计类编程题：

6. 计算机自动生成0-9之间的整数四则运算（+ - * /）计算题；由用户输入结果，然后自动批改结果是否正确。（采用生成随机数方式，百度rand()函数）、（/时只考虑整除）。

例：自动生成 3+9= 12(用户输入结果后回车),

显示： Correct!

提示：假设自动生成的整数四则运算为 data1 op data2

- (1) 百度rand()函数，注意在程序中包含其头文件；
- (2) 利用 rand()函数生成 2 个 0~9 的整数，分别赋值给 data1 和 data2；
- (3) 利用 rand()函数生成 1 个 0~3 的整数赋值给变量 c，表示四则运算中的运算符 op；
- (4) 若 c=0 则 op='+'、若 c=1 则 op='-','若 c=2 则 op='*'、若 c=3 则 op='/'（可利用 switch-case 结构）。在 switch-case 结构中，还需计算运算结果 result=data1 op data2；
- (5) 输出运算式 data1 op data2=, 输入用户的结果 guess。根据 result 和 guess 是否相等输出 "Correct!" 或 "Error!"。

7. 日期显示:输入今天的日期，输出明天的日期。

例：输入 2017-12-31,

输出 2018-01-01,

输入 2000-02-28,

输出 2000-02-29

提示：

- (1) 题目要求输入格式为 "year-month-day"，因此输入分隔符为 "-"；
- (2) 本题需要根据输入的 year 判断是否为闰年，方法详见教材 P18 例 2.3，可设置标志 flag 表示是否为闰年；
- (3) 特殊日期：
 - 第一种：“年-02-28”，若 year 为闰年下一天为“年-02-29”、否则下一天为“年-03-01”；
 - 第二种：“年-12-31”，则下一天为“年++-01-01”；
 - 第三种：1、3、5、7、8、10 月的 31 日，或 4、6、9、11 月的 30 日，则 month++、day=1；
 - 其它日期（非特殊日期）day++；
- (5) 题目要求月、日为 2 位输出宽度，不足 2 为左补 0。

综合编程题

8. 编程反复显示2010–2020年中选定的年份的月历。（2010年1月1日为星期五）。

例如：选定 2017 年，程序运行输出示例如下：

```

Enter which year you want to display: 2017
Enter month in 2017 to display calendar (1-12): 1
*****
  MON   TUE   WED   THU   FRI   SAT   SUN
                1
    2     3     4     5     6     7     8
    9    10    11    12    13    14    15
   16    17    18    19    20    21    22
   23    24    25    26    27    28    29
   30    31
Would you like to display another month (y/n) ? y
Enter month in 2017 to display calendar (1-12): 13
ERROR--- Enter month in range of (1-12)!
Enter month in 2017 to display calendar (1-12):9
*****
  MON   TUE   WED   THU   FRI   SAT   SUN
                1     2     3
    4     5     6     7     8     9    10
   11    12    13    14    15    16    17
   18    19    20    21    22    23    24
   25    26    27    28    29    30
Would you like to display another month (y/n) ? n

```

提示：如要输出2017年的日历（题目要求反复输出“选定年份”的日历，因此在程序运行过程中year值不变）

（1）需要的变量

①year、month——从键盘输入的年、月；

②fday——初始值为5表示2010年1月1日为星期5，之后用于表示year年month月1日为星期几；

③daysofyears——从2010年到year-1年（整年）的总天数，由于year不变，因此该值单独定义，在程序中只求一次值即可；

daysofthisyear——year年1月到month-1月（整月）的总天数，由于反复输入month，因此该值在循环过程中值会发生变化；

daysofyears+ daysofthisyear即为2010年到year年month-1月的总天数，可用于计算year年month月1日为星期几；

④数组int dayinmonth[12]，存放year年12个月每月的天数。

（2）输入year（在循环外）

①判断year是否为闰年（方法详见教材P18例2.3），设置2月天数dayinmonth[1]为29或28（if-else）；

②累加计算daysofyears，即2010年到year-1年的总天数（循环），需要判每一年是否为闰年（闰年则累加366，平年则累加365）；

(3) 循环, 根据输入的month循环输出日历, 直到用户输入的ch为n或N

①每次重置daysofthisyear=0; fday=5; 防止上一次循环对本次循环的影响;

②输入month, 使用“强壮的输入方式”保证输入的month在1-12之间;

③累加计算daysofthisyear, 即year年从1月到month-1月的总天数(循环), 从数组dayinmonth中取出各月的天数;

④daysofyears+daysofthisyear表示2010年到year年month月前的总天数, 再根据fday=5 (2010年1月1日为星期五), 计算year年month月1日为星期几

fday=(daysofyears+daysofthisyear+fday-1)%7; (对应示例中从周一开始输出, 自己思考如何得到该对应关系);

⑤输出year年month月的日历

首先, 按格式输出表头

```
printf("*****\n");
```

```
printf("    MON    TUE    WED    THU    FRI    SAT    SUN\n");
```

“MON”等前设置4个空格, 后续日期可使用%7d输出实现右对齐;

其次, 输出fday套(7个/套, 与%7d一致)空格;

最后, 输出month月的各日期, month的天数存放于数组元素dayinmonth[month-1]中(如3月的天数dayinmonth[2]=31), 用循环从1开始输出该月的所有日期, 注意每行输出周日后换行;

⑥接收用户输入, n或N表示结束(跳出循环); 否则继续输出日历。注意字符输入之前清空输入缓冲区。

程序大体框架如下:

```
#include <stdio.h>
main()
{
    int year, month, daysofyears=0, daysofthisyear=0;
    //daysofyears表示2010年到year年之前的(整年)的总天数
    //daysofthisyear表示year年前month-1个月的总天数
    //daysofyears+daysofthisyear表示2010年到year年month月前的总天数, 用于计算year年month月1日为星期几
    int fday=5; //已知2010年1月1日为星期五
    int dayinmonth[12]={31,28,31,30,31,30,31,31,30,31,30,31};
    //存放的year12个月对应的天数, 根据是否是闰年修改2月天数
    int i;
    char ch; //接收用户输入, y为继续输出日历, n或N为结束
    //输入年份year (选定年份, 根据题意位于循环外)

    //判断year是否为闰年, 设置2月天数dayinmonth[1]为29或28 (if-else)

    //累加计算daysofyears, 即2010年到year-1年的总天数(循环), 需要判断某年是否为闰年 (闰年则累加366, 平年则累加365)

    while(1) //根据输入的month循环输出日历, 直到用户输入的ch为n或N
    {
        daysofthisyear=0; //每次循环重置初始值
        fday=5;
        //输入month, 使用“强壮的输入方式”保证输入的month在1-12之间

        //累加计算daysofthisyear, 即year年从1月到month-1月的总天数(循环), 从数组dayinmonth中取出各月的天数

        //根据fday=5 (2010年1月1日为周五), 及daysofyears+daysofthisyear (2010年到year年month月前的总天数),
        //计算year年month月1日为周几

        //按格式输出日历表头

        //根据计算的fday (year年month月1日为周几)输出fday套(7个/套)空格, 循环

        //从数组元素dayinmonth[month-1]中取出该月天数, 循环输出该月所有日期 (使用%7d输出右对齐), 注意换行

        //接收用户输入, n或N表示结束; 否则继续输出日历。注意字符输入之前清空输入缓冲区
    }
}
```